

METHOD AND APPARATUS FOR CONTROLLING LOOP FILTERING OR POST
FILTERING IN BLOCK BASED MOTION COMPENSATIONED VIDEO CODING

10

BACKGROUND

Block based motion compensated video coding is used in many video
compression standards such as H.261, H.263, H263+, MPEG-1, MPEG-2, and H26L. The
lossy compression process can create visual artifacts in the decoded images, referred to as
image artifacts. Blocking artifacts occur along the block boundaries in an image and are
caused by the coarse quantization of transform coefficients.

Image filtering techniques can be used to reduce artifacts in reconstructed images.
Reconstructed images are the images produced after being inverse transformed and decoded.
The rule of thumb in these techniques is that image edges should be preserved while the rest
of the image is smoothed. Low pass filters are carefully chosen based on the characteristic of
a particular pixel or set of pixels surrounding the image edges.

Non-correlated image pixels that extend across image block boundaries are
specifically filtered to reduce blocking artifacts. However, this filtering can introduce
blurring artifacts into the image. If there are little or no blocking artifacts between adjacent
blocks, then low pass filtering needlessly incorporates blurring into the image while at the
same time wasting processing resources.

The present invention addresses this and other problems associated with the prior art.

SUMMARY OF THE INVENTION

Adjacent blocks are identified in an image. Coding parameters for the adjacent blocks are identified. Deblock filtering between the identified adjacent blocks is skipped if the coding parameters for the identified adjacent blocks are similar and not skipped if the coding parameters for the identified adjacent blocks are substantially different.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a diagram showing how deblock filtering is selectively skipped according to similarities between adjacent image blocks.

FIG. 2 is a diagram showing two adjacent image blocks having similar motion vectors.

FIG. 3 is a diagram showing how transform coefficients are identified for one of the image blocks.

FIG. 4 is a diagram showing how residual transform coefficients are compared between two adjacent image blocks.

FIG. 5 is a block diagram showing how the video image is encoded and decoded.

FIG. 6 is a block diagram showing how deblock filtering is selectively skipped in a codec.

FIG. 7 shows a table containing the results from selective deblock filter skipping.

DETAILED DESCRIPTION

In conventional filtering methods, filter processing only considers a single reconstructed image frame at a time. The motion-vector information available at both the encoder and decoder is not used. If two adjacent blocks share the same motion vector with

5 respect to the same reference image frame, (for a multiple reference frames system) there
may be no significant difference between the image residuals of each block. The block
boundary of these two adjacent blocks may have been filtered in the reference frame and
should therefore not be filtered again for the current frame. If a deblock filter is used without
considering this motion-vector information, the conventional filtering process might filter the
10 same boundary again and again from frame to frame. This unnecessary filtering not only
causes unnecessary blurring but also results in extra filter computations.

FIG. 1 shows an image 12 that selectively filters block artifacts according to
similarities between image blocks. The image 12 includes multiple image blocks 14 that are
each individually encoded before being stored, transmitted, etc. The borders between some
15 of the blocks 14 include blocking artifacts 18. Blocking artifacts are any image
discontinuities between blocks 14 that may be created by the encoding process. A low pass
filter is used to reduce the blocking artifacts that exist at the borders of adjacent image
blocks.

For example, blocking artifacts 24 exist between blocks 20 and 22. A low pass filter
20 is used at the border 26 between blocks 20 and 22 to remove or reduce the blocking artifacts
24. The low pass filter in one example selects a group of pixels 28 from both sides of the
border 26. An average pixel value is derived from the group of pixels 28. Then each
individual pixel is compared to the average pixel value. Any pixels in group 28 outside of a
predetermined range of the average pixel value is then replaced with the average pixel value.

25 As described above, if there are little or no blocking artifacts 24 between the adjacent
pixels, then the group of pixels 28 may be needlessly filtered causing blurring in the image
and wasting processing resources. A skip mode filtering scheme uses the motion estimation
and compensation information for adjacent image blocks. If the motion estimation and

5 compensation information is similar, deblock filtering is skipped. This not only avoids unnecessary image blurring but also significantly reduces the required number of filtering operations.

For example, it is determined during the encoding process that adjacent image blocks 30 and 32 have similar coding parameters. Accordingly, deblock filtering is skipped for the
10 groups of pixels 34 that extend across the border 31 between adjacent blocks 30 and 32. Skip mode filtering can be used for any horizontal or vertical boundary between any adjacent blocks in image 12.

FIG. 2 shows reference frames 42 and 48 and a current frame 40 that is currently being encoded or decoded. Coding parameters for blocks 44 and 46 are compared to
15 determine whether deblock filtering should be skipped between the two adjacent blocks 44 and 46. One encoding parameter that is compared is the Motion Vectors (MV) for the blocks 44 and 46.

The motion vector MV1 points from block 44 in current image frame 40 to an associated block 44' in the reference image 42. The motion vector MV2 points from block 46
20 in current image frame 40 to an associated block 46' in reference frame 42. Skip mode filtering checks to see if the motion vectors MV1 and MV2 point to adjacent blocks in the same reference frame 42. If the motion vectors point to adjacent blocks in the reference frame (MV1=MV2), then deblock filtering is skipped. This motion vector information may be used along with other coding information to decide whether to skip deblock filtering
25 between the two image blocks 44 and 46.

More than one reference frame may be used during the encoding and decoding process. For example, there may be another reference frame 48. The adjacent blocks 44 and 46 may have motion vectors pointing to different reference frames. In one embodiment, the

5 decision to skip deblock filtering depends on whether the motion vectors for the two adjacent blocks point to the same reference frame. For example, image block 44 may have a motion vector 49 pointing to reference frame 48 and image block 46 may have the motion vector MV2 pointing to reference frame 42. Deblock filtering is not skipped in this example because the motion vectors 49 and MV2 point to different reference frames.

10 FIG. 3 shows another coding parameter that may be used to decide whether or not to skip deblock filtering. The image block 44 from image frame 40 is compared with reference block 44' from the reference frame 42 pointed to by the motion vector MV1 as previously shown in FIG. 2. A residual block 44" is output from the comparison between image block 44 and reference block 44'. A transform 50 is performed on the residual block 44" creating a transformed block 44''' of transform coefficients. In one example, the transform 50 is a
15 Discrete Cosine Transform. The transformed block 44''' includes a D.C. component 52 and A.C. components 53.

The D.C. component 52 refers to a lowest frequency transform coefficient in image block 44. For example, the coefficient that represents the average energy in the image block
20 44. The A.C. components 53 refer to the transform coefficients that represent the higher frequency components in the image block 44. For example, the transform coefficients that represent the large energy differences between pixels in the image block 44.

FIG. 4 shows the transformed residual blocks 44''' and 46'''. The D.C. components 52 from the two transformed blocks 44''' and 46''' are compared in processor 54. If the D.C.
25 components are the same or within some range of each other, the processor 54 notifies a deblock filter operation 56 to skip deblock filtering between the border of the two adjacent blocks 44 and 46. If the D.C. components 52 are not similar, then no skip notification is initiated and the border between blocks 44 and 46 is deblock filtered.

5 In one embodiment, skip mode filtering is incorporated into the Telecommunications Sector of the International Telecommunication Union (ITU-T) proposed H.26L encoding scheme. The H.26L scheme only uses 4x4 integer Discrete Cosine Transform (DCT) blocks. Here, only the D.C. component of the two adjacent blocks may be checked. However some limited low frequency A.C. coefficients could also be checked when the image blocks are
10 bigger sizes, such as 8x8 or 16x16 blocks. For example, the upper D.C. component 52 and the three lower frequency A.C. transform coefficients 53 for block 44" may be compared with the upper D.C. component 52 and three lower frequency A.C. transform coefficients 53 for block 46". Different combinations of D.C. and/or low frequency A.C. transform coefficients can be used to identify the relative similarity between the two adjacent blocks 44
15 and 46.

The processor 54 can also receive other coding parameters 55 that are generated during the coding process. These coding parameters include the motion vectors and reference frame information for the adjacent blocks 44 and 46 as described above. The processor 54 uses all of these coding parameters to determine whether or not to skip deblock filtering
20 between adjacent image blocks 44 and 46. Other encoding and transform functions performed on the image may be carried out in the same processor 54 or in a different processing circuit. In the case where all or most of the coding is done in the same processor, the skip mode is simply enabled by setting a skip parameter in the filtering routine.

FIG. 5 shows how skip mode filtering is used in a block-based motion-compensated
25 Coder-Decoder (Codec) 60. The codec 60 is used for inter-frame coding. An input video block from the current frame is fed from box 62 into a comparator 64. The output of a frame buffering box 80 generates a reference block 81 according to the estimated motion vector (and possible reference frame number). The difference between the input video block and the

5 reference block 81 is transformed in box 66 and then quantized in box 68. The quantized transform block is encoded by a Variable Length Coder (VLC) in box 70 and then transmitted, stored, etc.

The encoding section of the codec 60 reconstructs the transformed and quantized image by first Inverse Quantizing (IQ) the transformed image in box 72. The inverse
10 quantized image is then inverse transformed in box 74 to generate a reconstructed residual image. This reconstructed residual block is then added in box 76 to the reference block 81 to generate a reconstructed image block. Generally the reconstructed image is loop filtered in box 78 to reduce blocking artifacts caused by the quantization and transform process. The filtered image is then buffered in box 80 to form reference frames. The frame buffering in
15 box 80 uses the reconstructed reference frames for motion estimation and compensation. The reference block 81 is compared to the input video block in comparator 64. An encoded image is output at node 71 from the encoding section and is then either stored or transmitted.

In a decoder portion of the codec 60, a variable length decoder (VLD) decodes the encoded image in box 82. The decoded image is inverse quantized in box 84 and inverse
20 transformed in box 86. The reconstructed residual image from box 86 is added in the summing box 88 to the reference block 91 before being loop filtered in box 90 to reduce blocking artifacts and buffered in box 92 as reference frames. The reference block 91 is generated from box 92 according to the received motion vector information. The loop filtered output from box 90 can optionally be post filtered in box 94 to further reduce image
25 artifacts before being displayed as a video image in box 96. The skip mode filtering scheme can be performed in any combination of the filtering functions in boxes 78, 90 and 94.

The motion estimation and compensation information available during video coding are used to determine when to skip deblock filtering in boxes 78, 90 and/or 94. Since these

5 coding parameters are already generated during the encoding and decoding process, there are no additional coding parameters that have to be generated or transmitted specially for skip mode filtering.

FIG. 6 shows is further detail how skip mode filtering is used in the filters 78, 90, and/or 94 in the encoder and decoder in FIG. 5. The interblock boundary between any two adjacent blocks "j" and "k" is first identified in box 100. The two blocks may be horizontally or vertically adjacent in the image frame. Decision box 102 compares the motion vector $mv(j)$ for block j with the motion vector $mv(k)$ for block k. It is first determined whether the two adjacent blocks j and k have the same motion vector pointing to the same reference frame. In other words, the motion vectors for the adjacent blocks point to adjacent blocks ($mv(j) = mv(k)$) in the same reference frame ($ref(j) = ref(k)$).

It is then determined whether the residual coefficients for the two adjacent blocks are similar. If there is no significant difference between the image residuals of the adjacent blocks, for example, the two blocks j and k have the same of similar D.C. component ($dc(j) = dc(k)$), then the deblock filtering process in box 104 is skipped. Skip mode filtering then moves to the next interblock boundary in box 106 and conducts the next comparison in decision box 102. Skip mode filtering can be performed for both horizontally adjacent blocks and vertically adjacent blocks.

In one embodiment, only the reference frame and motion vector information for the adjacent image blocks are used to determine block skipping. In another embodiment, only the D.C. and/or A.C. residual coefficients are used to determine block skipping. In another embodiment, the motion vector, reference frame and residual coefficients are all used to determine block skipping.

5 The skip mode filtering scheme can be applied to spatially sub-sampled chrominance channels. For example in a case with 4:2:0 color format sequences, skip mode filtering for block boundaries may only rely on the equality of motion vectors and D.C. components for the luminance component of the image. If the motion vectors and the D.C. components are the same, deblock filtering is skipped for both the luminance and chrominance components of the adjacent image blocks. In another embodiment, the motion vectors and the D.C. components are considered separately for each luminance and chrominance component of the adjacent blocks. In this case, a luminance or chrominance component for adjacent blocks may be deblock filtered while the other luminance or chrominance components for the same adjacent blocks are not deblock filtered.

10
15 FIG. 7 includes a table 110 showing the results of skip mode filtering using a ITU-TH.26L Testing Model-Long TML5.0. Table 110 compares the results of the TML filtering standard with skip mode filtering as described above. Encoding results using skip mode filtering are shown in table 110 under the heading SLA.

20 There were four images that were tested, Akiyo_cif for 300 frames at 30 Frames Per Second (fps), Foreman_cif for 300 frames at 30 fps, Foreman_qcif for 100 frames at 10 fps, and Tempete_cif for 260 frames at 30 fps. Quantization Parameters (QP) of 25 and 30 were used. The results show no significant visual quality degradation with the skip mode filtering. The Picture Signal to Noise Ratio (PSNR) for the images stays approximately the same for the luminance Y and chrominance U and V channels. However, skip mode filtering provides
25 time savings of 40-70 percent.

Skip mode filtering can be used with any system that encodes or decodes multiple image frames. For example, DVD players, video recorders, or any system that transmits

5 image data over a communications channel, such as over television channels or over the Internet.

The skip mode filtering described above can be implemented with dedicated processor systems, micro controllers, programmable logic devices, or microprocessors that perform some or all of the operations. Some of the operations described above may be
10 implemented in software and other operations may be implemented in hardware.

For the sake of convenience, the operations are described as various interconnected functional blocks or distinct software modules. This is not necessary, however, and there may be cases where these functional blocks or modules are equivalently aggregated into a single logic device, program or operation with unclear boundaries. In any event, the
15 functional blocks and software modules or described features can be implemented by themselves, or in combination with other operations in either hardware or software.

Having described and illustrated the principles of the invention in a preferred embodiment thereof, it should be apparent that the invention may be modified in arrangement and detail without departing from such principles. Claim is made to all modifications and
20 variation coming within the spirit and scope of the following claims.